

design ideas

Edited by Bill Travis and Anne Watson Swager

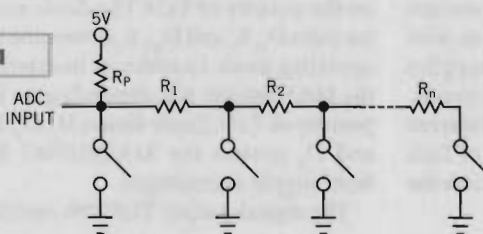
ADC circuit optimizes key encoding

Vitor Amorim, Siemens, and J Simões, University of Coimbra, Coimbra, Portugal

YOU CAN IMPLEMENT a simple and low-cost keyboard encoder by using a key-controlled resistive divider connected to an ADC (**Figure 1**). This type of circuit is especially suitable for embedded systems that use a μ C with an integrated ADC section. To obtain the best noise margin between keys, select resistors to yield an equal division of the voltage levels. To meet this criterion using the circuit in **Figure 1**, you must use a set of resistors with all-different, specific values. For example, a 10-key keyboard uses a 10-k Ω pullup resistor, R_p , and values of 1.1, 1.3, 1.8, 2.4, 3.3, 5.1, 8.2, 16, and 51 k Ω for R_1 through R_9 . Using commonly available resistor values and tolerances and taking account of the key resistivity and ADC linearity errors, you're limited in the number of keys you can encode with a safe noise margin. Using an 8-bit ADC, the cited 10-key example is close to that limit.

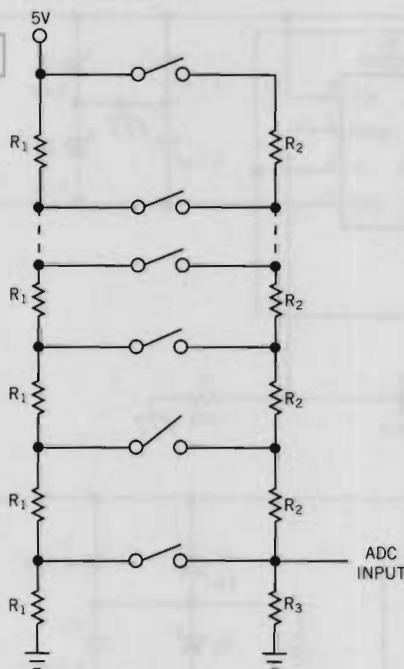
The circuit in **Figure 2**, despite its many resistors, overcomes the problem of the many restore values. The circuit is symmetric and uses the same two resis-

Figure 1



This is a simple way to encode keys, but it requires a range of resistor values for a large number of keys.

Figure 2



This key-encoding circuit simplifies resistor inventories, and provides a comfortable noise margin to boot.

tor values for all keys. It is also easy to expand the circuit for more keys. As in **Figure 1**'s circuit, if you simultaneously press more than one key, the circuit detects only the key whose connection is closest to the ADC. The chain of R_1 resistors de-

fines the voltage level associated with each key. Their nominal value is a trade-off between the power dissipation and the values of R_2 and R_3 , which depend on R_1 ; the total number of keys; and the desired noise margin. R_2 minimizes the voltage deviations at the nodes of the R_1 chain whenever you simultane-

ously press two or more keys. Therefore, you should calculate its value, much higher than that of R_1 , by taking into account R_1 and the desired width of the voltage window associated with each key. Similarly, R_3 's value should be much higher than that of R_2 to ensure that the voltage level associated with each key almost completely transfers to the ADC's input.

The circuit was tested by encoding 15 keys with one 8-bit analog input of the Microchip PIC16C71 μ C. The resistor values are 47 Ω , 3.9 k Ω , and 4.7 M Ω for R_1 , R_2 , and R_3 , respectively. Only R_1 needs a tight tolerance; the others are less demanding. The window of key acceptance is set to a 7-LSB interval centered on the theoretical key level. This interval is large enough to accommodate the worst case (simultaneously pressing the two more-distant keys from the analog input) with a safety noise margin between valid keys of 10 LSBs. (DI #2319).

TO VOTE FOR THIS DESIGN,
CIRCLE NO. 332

ADC circuit optimizes key encoding	101
RS-232C circuit has galvanic isolation.....	102
Digital pot adjusts LCD's contrast.....	104
Add speech encoding/decoding to your design	106
Cheap PWM IC makes synchronous gate driver.....	110
Circuit detects total on-time	112

RS-232C circuit has galvanic isolation

Ioan Ciascai, REI Data, Napoca, Romania

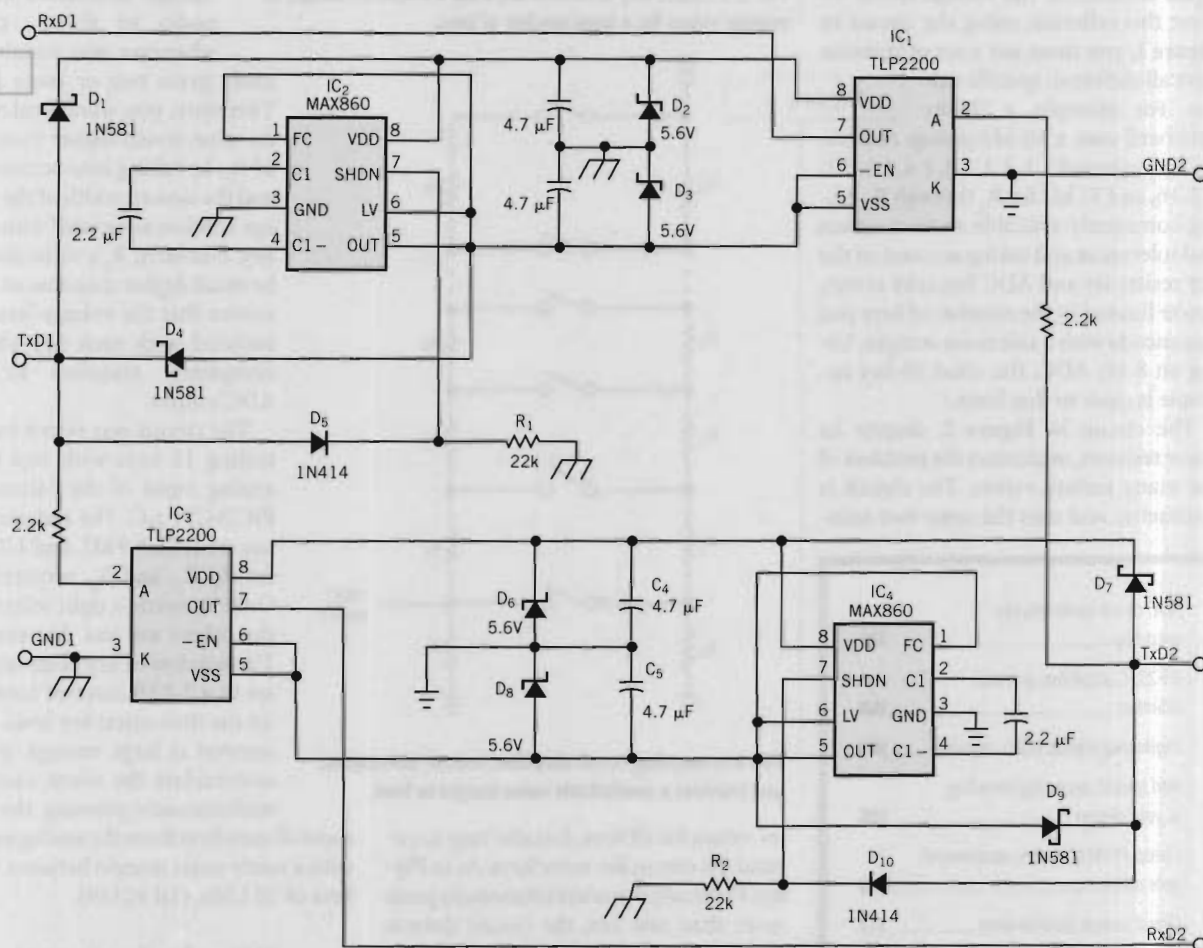
YOU CAN OBTAIN LONGER TRANSMISSION distances with the RS-232C interface if you use galvanic isolation between the two linked terminals. Galvanic isolation also eliminates problems arising from disparate potentials between terminals. Using two MAX860/861 ICs and two TPL2200 optoisolators, you can obtain galvanic isolation for three-wire transmission without external supplies (Figure 1). The MAX860/861 circuits, which generate two voltages of different polarity, regardless of the polarity of TxD, which provides the supply voltage, are the basis of the design.

For the positive TxD polarity, the MAX860/861 ICs function as voltage inverters, whereas for negative-polarity TxD, the ICs function as voltage doublers. Diodes D_1 and D_7 provide the positive supply voltages; D_4 and D_9 provide the negative supply voltages, depending on the polarity of TxD. The diode-resistor pairs D_5, R_1 and D_{10}, R_2 determine the operating mode (doubler or inverter) of the MAX860/861 ICs, depending on the polarity of TxD. Zener diodes D_2, D_3, D_6 , and D_8 protect the MAX860/861 ICs from supply overvoltages.

The digital-output TPL2200 optoisolators provide galvanic isolation and generate the RxD received signals with RS-232C logic levels. The circuit provides galvanically isolated communication in full-duplex mode at any standard transmission speed. The use of galvanic isolation allows transmission over nearly twice the distance for nonisolated systems. If you use a four-wire cable and split the separation circuits at the cable ends, you can further increase the transmission distance. (DI #2315).

TO VOTE FOR THIS DESIGN,
CIRCLE NO. 333

Figure 1



RS-232C-interface ICs and optoisolators provide a supplyless RS-232C transmission link with galvanic isolation for increased distance.

Digital pot adjusts LCD's contrast

Jef Collin, CSE Systems, Turnhout, Belgium

YOU CAN USE A DIGITALLY controlled potentiometer for many purposes. In this example, you can use the device to regulate the contrast of a standard (such as two lines by 40 characters) LCD. You can use the circuit in **Figure 1** in a portable test system, in which you need to change the contrast of the LCD as a function of the viewing angle. You choose the contrast setup from a menu and then use up or down buttons with μC IC₁ to adjust the contrast. The μC stores the contrast value in the digital potentiometer, IC₂. This design uses a Xicor 10-k Ω unit (dubbed "EEPOT"), but you could use other devices in the design.

The EEPOT connects to the LCD's VO line. You could connect the other side of the potentiometer to ground, but, for better contrast, you can apply a negative voltage to this terminal. This circuit has a

LISTING 1—ROUTINE FOR LCD-CONTRAST ADJUSTMENT

```

; *** BIT DEFINITIONS ***
EEPOT_INC BIT P1.4      ; EEPOT CONNECTIONS
EEPOT_UD BIT P1.5
EEPOT_CS BIT P1.6

; *** INITIAL SETTINGS ***
SETB EEPOT_CS
SETB EEPOT_INC
SETB EEPOT_UD

; *** UP CONTRAST ***
CONT_UP:  NOP
          SETB EEPOT_UD
          SETB EEPOT_INC
          NOP
          CLR EEPOT_CS
          NOP
          CLR EEPOT_INC
          NOP
          SETB EEPOT_INC
          NOP
          CLR EEPOT_INC
          NOP
          SETB EEPOT_INC
          NOP
          CLR EEPOT_INC

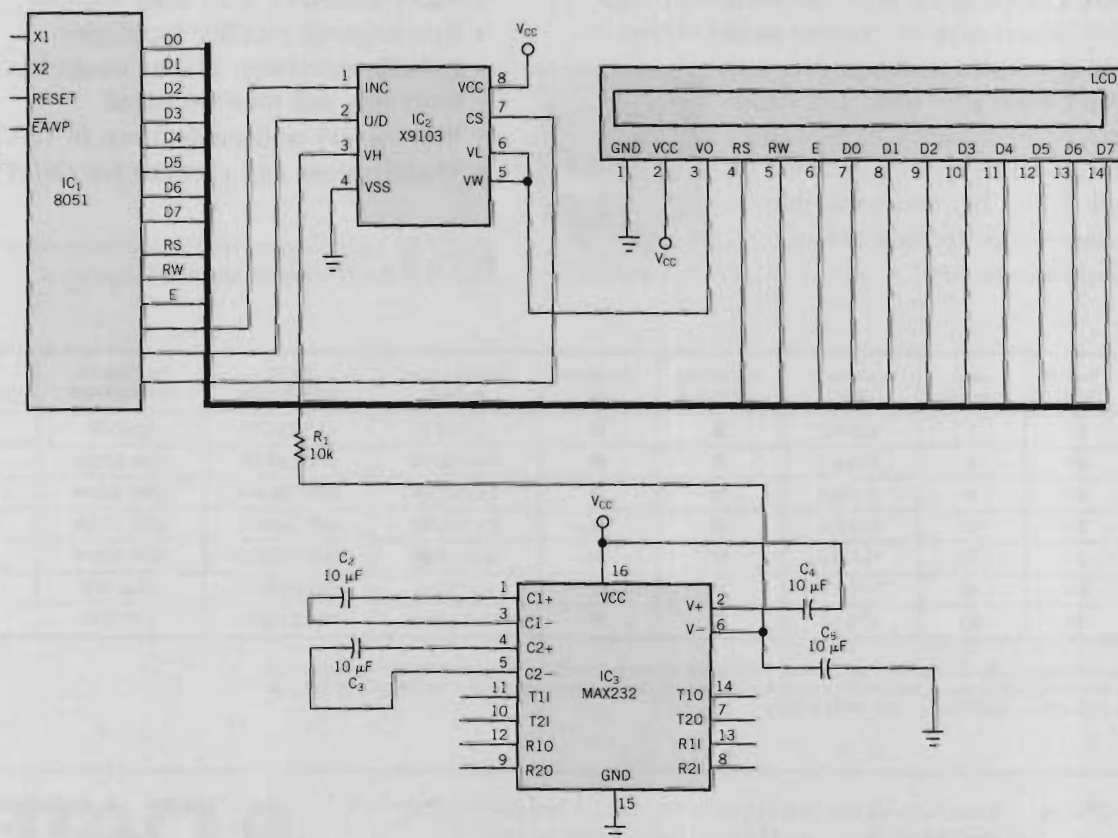
; *** DOWN CONTRAST ***
CONT_DWN: NOP
          CLR EEPOT_UD
          SETB EEPOT_INC
          NOP
          CLR EEPOT_CS
          NOP
          CLR EEPOT_INC
          NOP
          SETB EEPOT_INC
          NOP
          CLR EEPOT_INC
          NOP
          SETB EEPOT_INC
          NOP
          SETB EEPOT_CS
          RET
    
```

serial interface, so it uses the negative-voltage generator in the MAX232 RS-232C/TTL converter. **Listing 1** is the subroutine for the 8051 μC . The program

calls the routines for contrast-up or -down. (DI #2314).

TO VOTE FOR THIS DESIGN,
CIRCLE No. 334

Figure 1



A standard 8051 μC and a digitally controlled potentiometer provide a convenient way to vary the contrast of an LCD, using contrast-up and -down buttons.

Add speech encoding/decoding to your design

Rodger Richey, Microchip Technology Inc, Chandler, AZ

ADDING SPEECH capabilities to a design can sometimes lead to complex algorithms and expensive DSPs or specialized audio chips. However, with the completion of a simplified adaptive differential pulse-code-modulation (ADPCM) algorithm, you can now implement these audio capabilities in low-cost 8-bit μ Cs, which typically have lower power consumption and cost than their DSP or audio-chip counterparts. A two-chip design is feasible by offloading the encoding and decoding tasks onto the μ C as if it were a peripheral.

Since 1991, the Interactive Multimedia Association (IMA) Digital Audio Technical Working Group (DATWG) has been working to define a cross-platform digital-audio exchange format. An inherent problem exists with the exchange of audio data between PC, Mac, and workstation computers. Each computer has its own data types and sampling rates. In May 1992, IMA DATWG published the first revision of the Cross-Platform Digital Audio Interchange recommendation that specifies three uncompressed and one compressed data type at various sample rates. The compressed data type is the Intel (www.intel.com) 4-bit DVI ADPCM algorithm. This algorithm compresses a 16-bit signed audio sample into 4 bits and takes advantage of the high correlation between consecutive samples, which enables the prediction of future samples. Instead of encoding the sample itself, ADPCM encodes the difference between a predicted sample and the actual sample. This method provides more efficient

compression with fewer bits per sample and yet preserves the overall quality of the audio signal. Both the encoder routine (Listing 1) and the decoder routine (Listing 2) are written in C to ease readability.

LISTING 1—ADPCM ENCODER

```
/* Table of index changes */
const char IndexTable[16] = {-1,-1,-1,-1,2,4,6,8,-1,-1,-1,-1,2,4,6,8};

/* Quantizer step size lookup table */
const long StepSizeTable[89] = {7,8,9,10,11,12,13,14,16,17,19,21,23,25,28,31,34,37,41,
45,50,55,60,66,73,80,88,97,107,118,130,143,157,173,190,
209,230,253,279,307,337,371,408,449,494,544,598,658,724,
796,876,963,1060,1166,1282,1411,1552,1707,1878,2066,2272,
2499,2749,3024,3327,3660,4026,4428,4871,5358,5894,6484,
7132,7845,8630,9493,10442,11487,12635,13899,15289,16818,
18500,20350,22385,24623,27086,29794,32767};

signed int diff;          // Diff. between sample and predicted sample
int step;                 // Quantizer step size
signed int predsamp; diffq; // ADPCM predictor output, Predicted diff.
char index;               // Index into step size table

char ADPCMEncoder( signed int sample )
{
    char code;             // ADPCM output value

    predsamp = state.prevsamp; // Restore previous values of predicted
    index = state.previndex;  // sample and quantizer step size index
    step = StepSizeTable[index];

    diff = sample - predsamp; // Compute diff. between actual sample
    if(diff >= 0)             // and the predicted sample
        code = 0;
    else
    {
        code = 8;
        diff = -diff;
    }

    diffq = step >> 3;        // Quantize the diff. into 4-bit ADPCM code
    if( diff >= step )        // using the quantizer step size
    {
        code |= 4;           // Inverse quantize the ADPCM code into a
        diff -= step;        // predicted diff. using the quantizer step
        diffq += step;       // size
    }
    step >>= 1;
    if( diff >= step )
    {
        code |= 2;
        diff -= step;
        diffq += step;
    }
    step >>= 1;
    if( diff >= step )
    {
        code |= 1;
        diffq += step;
    }

    if( code & 8 )            // Compute new predicted sample by adding
        predsamp += diffq;   // the old predicted sample to new diff.
    else
        predsamp -= diffq;
    if( predsamp > 32767 )    // Check if overflow of the new pred. sample
        predsamp = 32767;
    else if( predsamp < -32768 )
        predsamp = -32768;

    index += IndexTable[code]; // Find new quantizer stepsize
    if( index < 0 )           // Check if overflow of new quantizer step
        index = 0;
    if( index > 88 )
        index = 88;

    state.prevsamp = predsamp; // Save values for next iteration
    state.previndex = index;
    return ( code & 0x0f );    // Return the ADPCM code
}
```

The hardware implementation depends on the type of interface. With a parallel interface, you can use the PIC16C556A (Microchip Technology, www.microchip.com) (Figure 1a). A standard parallel interface uses the chip-

select (CS), output-enable (OE), and write-enable (WR) pins on Port A. The 8-bit data interface connects to the 8-bit Port B on the μ C. You can use two additional I/O lines for status information, such as encode/decode select, to the μ C or an interrupt line to the main controller to indicate when data is ready. CS, which connects to the RA₄ pin, can interrupt the PIC16C556A on the start of a transmission.

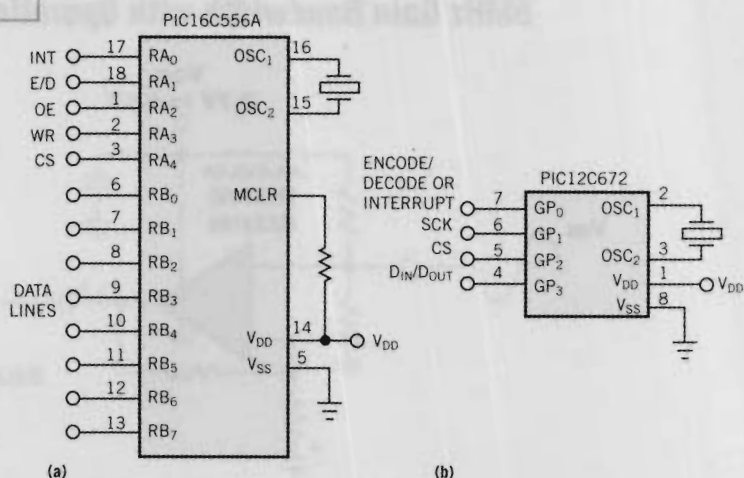
The second hardware implementation uses a serial interface and an eight-pin μ C (Figure 1b). The PIC12C672 uses four of the pins for power, ground, and oscillator input and output. Three of the remaining I/O pins are for clock (SCK), data in/out (D_{IN}/D_{OUT}), and CS. You can use the other I/O pin to indicate the desired encode/decode operation or as an interrupt to the main controller. CS connects to the external interrupt pin, GP₂, to detect the start of a transmission.

Both μ Cs have a flexible oscillator structure for use with a crystal, a resonator, or an external clock signal. Both parts of the figure show the μ C using an external crystal as the clock source. Because these devices are fully static, the main controller can provide the clock source to the μ C. By turning the clock source on and off as necessary, the main controller can further decrease overall power consumption. This method also allows control of the speed at which the algorithm runs, which is proportional to the sample rate of the system.

To fully implement the μ C as an ADPCM-encoder/decoder peripheral requires firmware to implement the serial or parallel interface, such as listings 1 and 2, and a main routine to tie everything together. The main controller is responsible for sampling the incoming audio waveform, storing and retrieving the ADPCM codes from nonvolatile memory, and then playing the resulting samples. The main controller feeds samples or ADPCM codes to the μ C and then reads the resulting ADPCM codes or samples from the μ C.

The listings are available for down-

Figure 1



Using the simplified ADPCM algorithm, you can now implement audio capabilities in μ Cs by offloading the encoding and decoding tasks onto the parallel (a) or serial (b) μ C as if it were a peripheral.

LISTING 2—ADPCM DECODER

```
// This routine also uses the IndexTable and StepSizeTable from Listing 1
signed int ADPCMDecoder( char code )
{
    predsamp = state.prevsamp; // Restore previous values
    index = state.previndex;

    step = StepSizeTable[index]; // Find quantizer step size

    diffq = step >> 3; // Inverse quantize ADPCM code into a
    if( code & 4 ) // diff. using the quantizer step
        diffq += step;
    if( code & 2 )
        diffq += step >> 1;
    if( code & 1 )
        diffq += step >> 2;

    if( code & 8 ) // Add the difference to predicted sample
        predsamp -= diffq;
    else
        predsamp += diffq;

    if( predsamp > 32767 ) // Check if overflow of new predicted sample
        predsamp = 32767;
    else if( predsamp < -32768 )
        predsamp = -32768;

    index += IndexTable[code]; // Find new quantizer step size
    if( index < 0 ) // Check if overflow of new quantizer step
        index = 0;
    if( index > 88 )
        index = 88;

    state.prevsamp = predsamp; // Save values for next iteration
    state.previndex = index;

    return( predsamp ); // Return new sample
}
```

loading from at EDN's Web site, www.ednmag.com. At the registered-user area, go into the Software Center to download the file from DI-SIG, #2292. (DI #2292)

TO VOTE FOR THIS DESIGN,
CIRCLE NO. 335

Cheap PWM IC makes synchronous gate driver

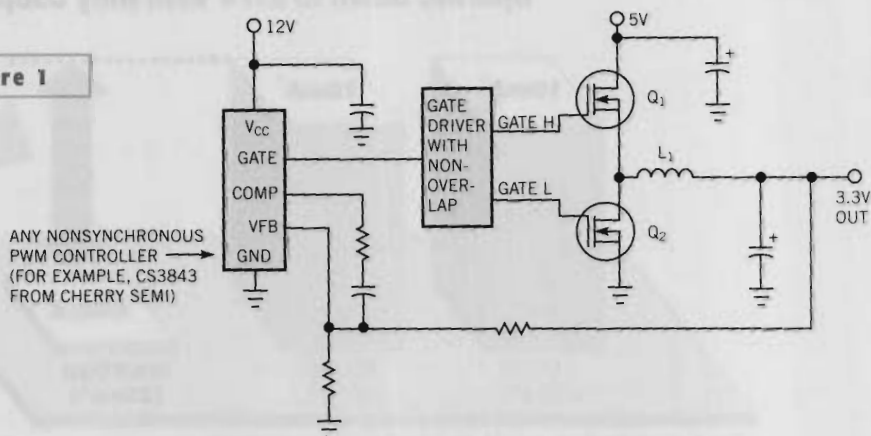
Dimitry Goder, Switch Power Inc, Campbell, CA

A SYSTEM WITH A μ P, memory, and peripherals usually requires several power-supply voltages. Designers typically use local switching regulators to produce the desired voltage rails. One of the most common topologies, the synchronous buck regulator, converts a 5 or 12V bus to some other, lower voltage. This approach has gained vast acceptance, thanks to its relative simplicity and high conversion efficiency. Specialized synchronous buck controllers are available, but you generally pay a premium for these ICs. Meanwhile, many inexpensive, general-purpose PWM controllers are available, but they require you to implement synchronous rectification with discrete circuitry. The implementation is a difficult task, because it involves building two out-of-phase, mutually exclusive gate drivers. The block diagram in **Figure 1** shows how to use any general-purpose PWM controller (for example, the CS3843 from Cherry Semiconductor in East Greenwich, RI) to configure a synchronous buck regulator.

The most critical aspect of the driver design is nonoverlap timing, which prevents simultaneous high states at gates H and L, thus eliminating the simultaneous turn-on of Q_1 and Q_2 , with resulting shoot-through currents. **Figure 2** shows details of the driver block. The input signal passes through two inverters, IC_{1A} and IC_{1B} , to generate the Gate H signal in phase with the input. The complementary follower, Q_1 and Q_2 , forms a current amplifier to provide sufficient drive for the top MOSFET. Meanwhile, IC_{1C} inverts the input, and Q_3 and Q_4 amplify the signal to drive the bottom MOSFET.

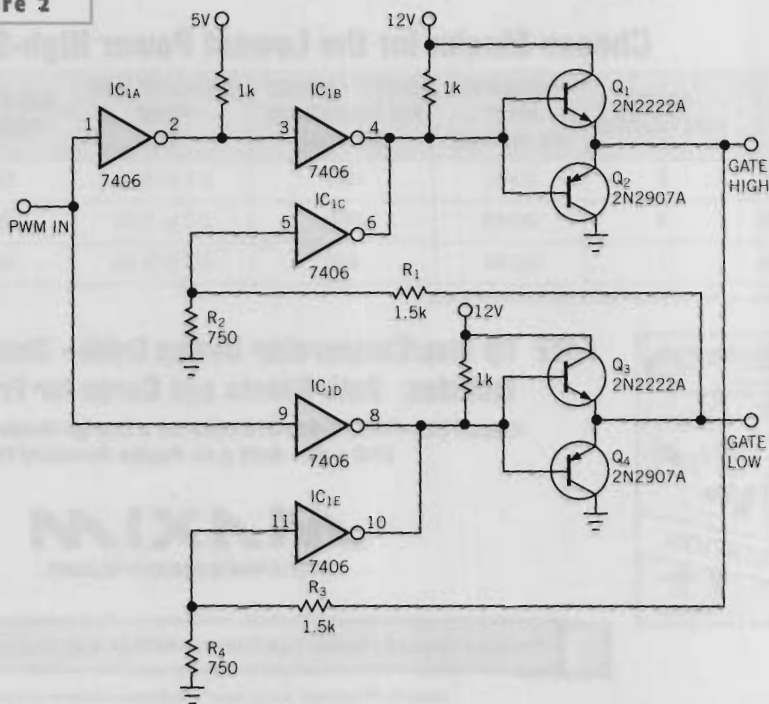
IC_{1C} and IC_{1E} provide the nonoverlap function. Each of the inverters ensures a low state at the corresponding gate-drive output until the other driver's output falls below a certain threshold. This

Figure 1



All it takes to drive high- and low-side MOSFETs from a PWM controller is a simple driver with 180° out-of-phase outputs, but you must beware of simultaneous MOSFET conduction.

Figure 2

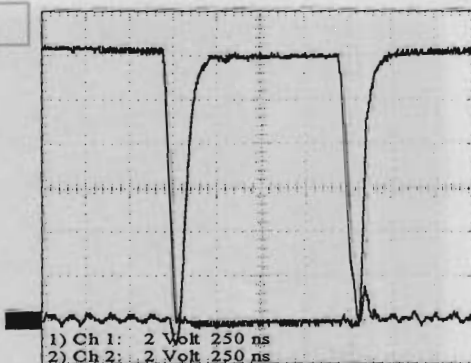


A sprinkling of TTL gates and four bipolar transistors provide efficient, nonoverlap drive to the two MOSFETs in a synchronous rectifier.

function prevents simultaneous high states, regardless of the circuit delays and the type of MOSFETs you use. The TTL switching threshold and the values of R_1 , R_2 and R_3 , R_4 determine the nonoverlap threshold. The threshold is approximately 3V for the circuit in **Figure 2**, but you can easily adjust it for other values.

Creating a high-side driver requires a bias voltage higher than the input voltage. A 12V line is commonly available, so you can use it to power the driver. The outputs of IC_{1B} and IC_{1D} must swing between ground and 12V; you thus need a 7406 open-collector inverter with a

Figure 3



At least 60 nsec separates the turn-on drive from the upper and lower drivers, thus preventing simultaneous MOSFET conduction.

high-voltage capability. If 12V is unavailable, you can use charge-pump circuitry to double the input voltage. The driver shows excellent performance, with 60-nsec nonoverlap time, superior to that of many available ICs. **Figure 3** shows the two gate drivers' outputs, with each driver switching an IRL3103 n-channel MOSFET, a good choice for a 10A converter. The total cost of the driver does not exceed 30 cents. (DI #2306).

TO VOTE FOR THIS DESIGN,
CIRCLE NO. 336

Circuit detects total on-time

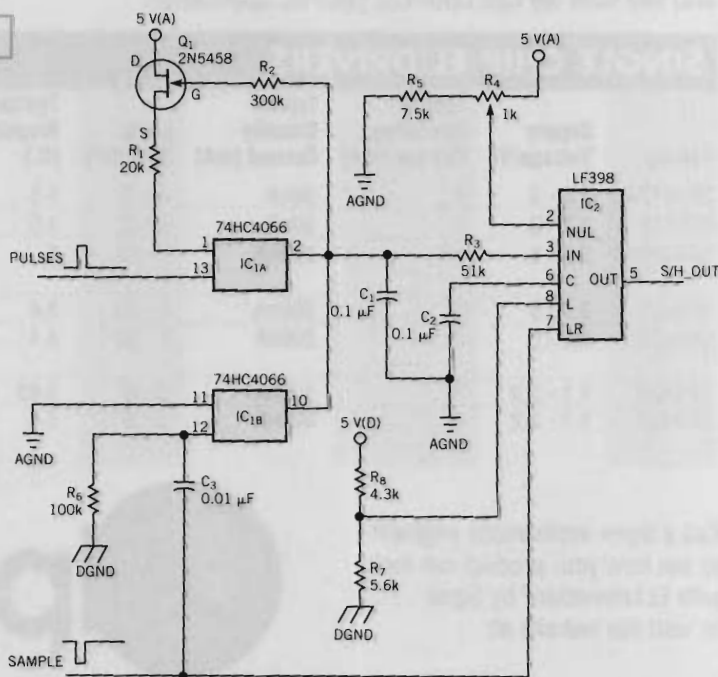
Richard Nachazel, Vista Electronics Co, Ramona, CA

THE CIRCUIT IN **Figure 1** provides a voltage that is directly proportional to the sum of the total on-times for a number of positive pulses that occur during the interval between sample signals. When a positive pulse arrives at the control input to switch IC_{1A} , a constant-current source charges capacitor C_1 through IC_{1A} for the duration of that pulse and for each subsequent pulse. The constant-current source comprises Q_1 , R_1 , C_1 , and R_2 , which drives Q_1 's gate at C_1 's potential. C_1 and C_2 should be noninductive and have low dielectric absorption.

When you apply a low signal to the Sample line, the sample/hold circuit (IC_2) latches the analog voltage on C_1 . You adjust R_4 for 0V at the output with no pulses at the input. The trailing edge of the Sample signal switches IC_{1B} on, discharging C_1 to an initial 0V potential. You can connect unused analog switches in parallel with IC_{1B} to speed the discharge. Higher supply voltages also improve the circuit's speed and performance. (DI #2300).

TO VOTE FOR THIS DESIGN,
CIRCLE NO. 337

Figure 1



Two ICs, a FET, and a handful of components form a pulse-width-to-voltage converter.

THANKS TO COST AND EFFICIENCY IMPROVEMENTS, THE SOLAR CELL IS BECOMING A VIABLE ALTERNATIVE TO TRADITIONAL POWER SOURCES FOR SOME APPLICATIONS. BUT DEVISING A RELIABLE SUN-BASED POWER SYSTEM GIVES NEW MEANING TO THE PHRASE "SAVING FOR A RAINY DAY." WITH KNOWLEDGE OF SOLAR POWER'S UNIQUE CHALLENGES, PLUS HELP FROM HIGHLY INTEGRATED DC/DC CONVERTERS, YOU CAN MAKE YOUR APPLICATION SHINE.

Harness solar power with smart power-conversion techniques

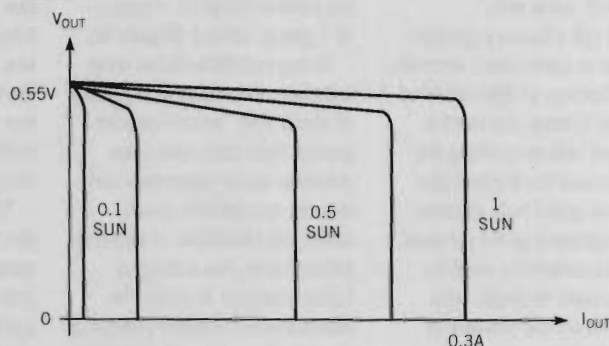
As a power source, the sun offers some impressive advantages over typical battery cells: It generates virtually limitless energy, requires no recharging, and is expected to last for another 5 billion years or so. Today's more efficient, less expensive solar cells provide a practical means of converting the sun's power into electricity to run our gadgets and gizmos (see sidebar "From photons to electrons"). The only clouds in this otherwise sunny picture are...well, clouds; because they often obscure the sun, it's tough to design a system that can reliably provide power.

The output current for typical monocrystalline-silicon solar cells directly depends on the amount of incident sunlight (Figure 1). For example, a typical hobby-grade credit-card-sized silicon cell has open-circuit voltage of 0.55V. Internal resistance causes a voltage drop as you draw current from the cell, but, as light energy drops below the level necessary to support the output load, the cell current-limits at an almost constant voltage. For a light level of one full sun (solar irradiance of 1 kW/m²), the cell provides a typical short-circuit current of 0.3A.

Maximum output power arises at the transition from constant voltage to constant current, typically 0.484V and 0.25 to 0.275A at one full sun. This

0.484V is too low for most applications, so solar panels usually connect the cells in a series/parallel combination that provides several amperes of current at about 12V. Such outputs are useful for many applications, assuming full sunlight is available whenever the application is active. Unfortunately, this imperative is hard to realize in most locations. Varying

Figure 1



Available current output from a solar cell is proportional to the incident-light energy, but the cell's open-circuit voltage is almost constant.

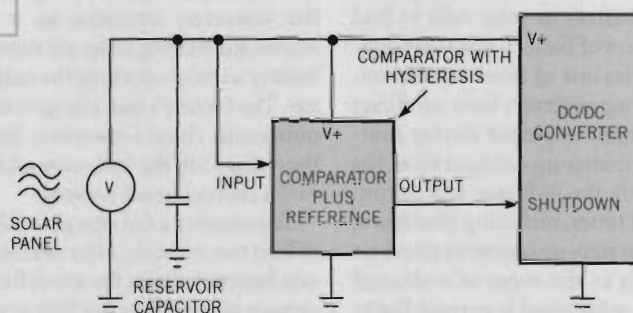
which then provides the load with a stable supply. Charging the battery directly from the solar panel (via a series diode), on the other hand, is difficult or at least inefficient, because battery voltage changes considerably as the battery charges.

Obtaining all the available energy from a solar panel requires a switch-mode step-up or -down converter and battery charger, in addition to the battery. The converter must ensure that every bit of energy taken from the solar panel is efficiently stored in the battery for future use. A switch-mode converter works in two cycles. It first connects an inductor to a power source, allowing a buildup of inductor current that stores energy in the inductor. In the second cycle, a change in current path enables the inductor to transfer its accumulated energy to the load. (The load voltage can be higher or lower than that of the inductor's power source.)

Connecting the inductor directly to a solar panel, however, is inefficient. Depending on light levels, the panel's output-current capability can range from microamps to several amperes. A much better approach is to connect a reservoir capacitor to the solar panel. By monitoring voltage on this capacitor, you can then turn on the switch-mode converter only when the panel output is optimum, that is, 0.484V times the number of (ranks of) series-connected cells in the panel. This input is adequate to start the converter, and the capacitor provides a low-impedance path for the inductor current. Moreover, the capacitor supports full cycles of converter operation without allowing the converter's input voltage to drop below its operating voltage. When the capacitor voltage drops below a predetermined level (the converter's minimum operating voltage or higher), the converter shuts down until the capacitor again charges to the optimum voltage.

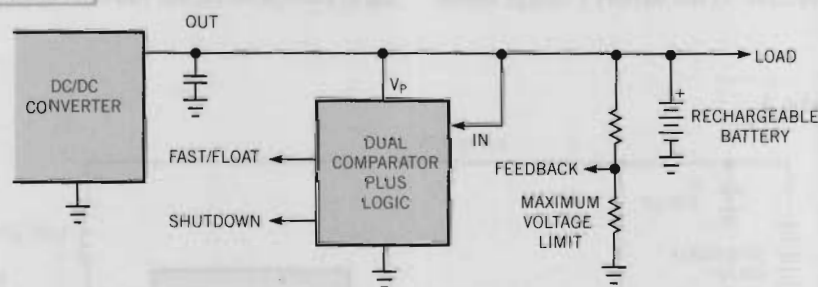
In **Figure 2**, a comparator with hysteresis controls the converter's shutdown pin. As long as the reservoir capacitor's voltage is lower than the optimum load voltage for the solar panel (0.484V times the series-connected-cell number), the converter is in shutdown. When the capacitor voltage reaches this optimum load voltage, the converter is enabled un-

Figure 2

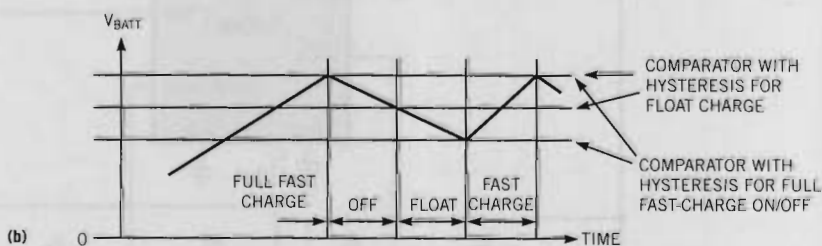


The input to a dc/dc converter for solar-panel power conversion comprises the panel, a reservoir capacitor, and a comparator/reference circuit that enables the converter only when adequate power is available.

Figure 3



(a)



(b)

The dc/dc converter enables a solar panel to charge a rechargeable battery (a). Comparator-generated signals enable the circuit to control the charging current (b).

til the capacitor voltage declines to the lower limit of the comparator's hysteresis band.

Thus, operating in bursts whenever there is sufficient charge on the capacitor, the converter charges the battery only when the value of reservoir-capacitor voltage is optimum for all levels of sunlight. In effect, the solar panel's load (the switch-mode converter) adjusts auto-

matically to the solar panel's best operating area (its V/I limit). The ratio of solar-panel no-load voltage to battery voltage determines whether the converter action must step up or step down.

If the solar-panel no-load voltage (determined by the number of series-connected cells) is lower than the voltage of the discharged battery, choose a step-up converter. If the panel voltage under

load is higher than that of the fully charged battery, choose a step-down converter. Otherwise, rearrange the number of battery or solar cells to find a combination of battery and solar panel that satisfies one of those conditions.

Step-down converters have no direct path from input to output during shutdown, but the step-up configuration has a dc path (via the inductor and output diode) at all times, including shutdown. Be aware that step-up converters have no current limit in the event of a shorted output. The solar panel is current-limited, so the main worry in such cases is probably a shorted battery, which may require other means of circuit protection.

THE BATTERY INTERFACE

An inspection of the converter's output side raises some other issues. The desired output voltage is not constant but depends on the battery's charge condi-

tion. When charging, you always want to apply a "fast charge" unless the battery is already fully charged. During fast charge, the converter operates as a current source, forwarding inductor energy to the battery without checking the output voltage. The battery's fast-charge-current requirement should therefore determine the values for the inductor and for the FET's current-sense resistor.

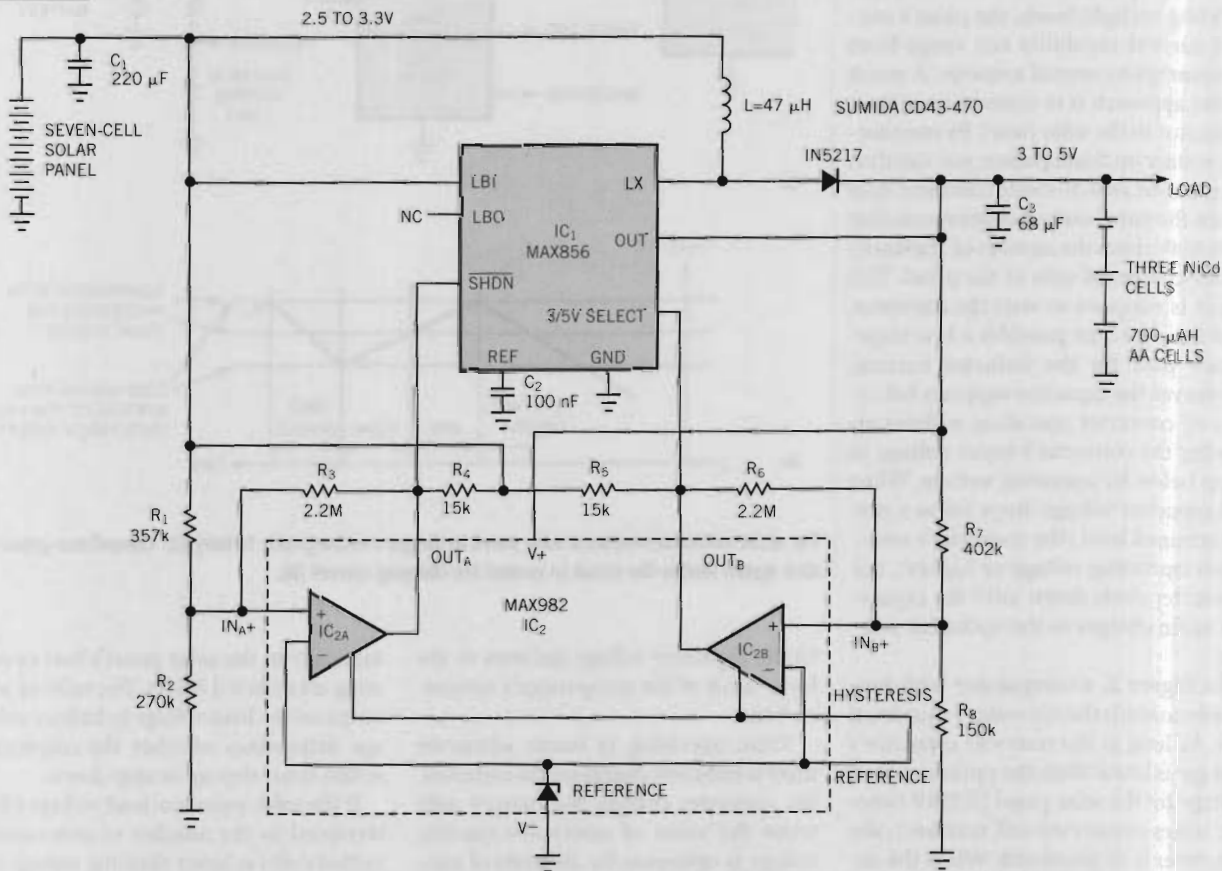
Terminating a fast charge is difficult for at least two reasons. After reaching a certain battery voltage, the above burst-conversion scheme does not hint at when the next conversion cycle will occur. If it occurs within milliseconds, you should probably stop the fast charge. If it occurs within hours or days (when sunlight is scarce), the fast charge should continue, because the battery may otherwise drain long before the next cycle. The other difficulty stems from charge current arriving in infrequent bursts: You can't detect

a zero or negative dV/dt under this condition.

One approach applicable to both lead-acid and NiCd batteries is to monitor the output voltage with another comparator, disabling switching when the voltage reaches a high limit and enabling it when the voltage declines below a predetermined level. A third level of detection could also enable a float-current (trickle) charge of the battery (Figure 3). When charging a discharged battery, the circuit applies full fast-charge current until the voltage reaches its upper limit. The circuit then disables switching until the voltage reaches the next-lowest (middle) limit, whereupon it enables the trickle charge. Trickle charging continues until the voltage reaches its upper limit (turning off the switching) or its lower limit (enabling the fast charge again).

A dual comparator with hysteresis controls the converter shutdown and the se-

Figure 4



This circuit enables a seven-cell solar panel to charge a three-cell NiCd battery.

lection of fast charge versus trickle charge. To prevent discharge of the battery when the solar-panel voltage goes to zero, you should design the comparator circuit not to load the battery via its supply terminal or the external resistive divider. When solar-panel voltage is low, you may control the converter shutdown from either the output comparator or the input comparator, and this capability may require additional logic.

The circuit in **Figure 4**, which charges a three-cell battery from a seven-cell solar panel, illustrates the above ideas. The battery voltage at discharge is greater than 3.0V, and the optimum solar-panel output is 2.9V (3.8V maximum), so the circuit requires a step-up dc/dc converter (IC₁). The dual comparator controls shutdown and charge termination. IC_{2A} directly controls shutdown, holding the converter in shutdown until the solar panel charges C₁ to 2.9V. The converter then becomes active, and hysteresis allows the operation to continue until the voltage drops below 2.5V. While on, the converter delivers its full output current, subject only to the internal current limit.

Regulation first begins when the battery voltage reaches 5.0V, but the output comparator (IC_{2B}) terminates the charge cycle at 4.6V. The design handles termination not by controlling the shutdown pin, which requires additional logic, but by the unconventional approach of driving the 3/5V-select pin high. Because the output is at 4.6V, selecting a 3V output (actually 3.3V) causes the converter to turn off. Again, comparator hysteresis ensures that the converter remains off until the battery voltage decreases to 4V. It then resumes operation, providing a simple form of idle charging that does not overcharge the battery.

Because the converter shuts down when the solar-panel output is low, the battery-discharge current (the sum of converter-shutdown current, comparator-supply current, R₇/R₈ current, and reverse current from the Schottky diode) is minimal. The Schottky reverse current is 5 μ A typical at 25°C and 50 μ A or higher at 50°C, which is too high for some applications. For those, you can substitute a switching diode, such as the 1N4148, at the expense of slightly lower

converter efficiency. As with all step-up converters, you should avoid taking the output lower than the input. In this case, an overload or shorted battery cell can cause an uncontrolled dc current flow from the solar panel through the inductor and diode to the output. In most cases, however, current limiting in the solar panel allows the battery to simply recharge to 3V without damage to the inductor.

Applications in which the solar-panel voltage is lower than the low-battery voltage require a step-up converter, and applications that need more current (or a peak current other than that determined by the fixed internal setting) require a step-up controller. Applications in which the solar-panel voltage is higher than the fully charged battery voltage demand a step-down converter. As an example of applications demanding a step-down converter, configuring the MAX797 controller as a current source and its internal 5V linear regulator can power the external comparators. The device is well-suited for charging a car battery from two 12V solar panels.

These circuits and techniques should ensure efficient battery charging from solar panels despite unpredictable weather conditions. The target applications are primarily low to medium power, from a few watts to 100W. A further refinement in the charger circuit, though not required for most NiCd and lead-acid batteries, could sense the battery current and use that information to provide a constant current-source output with voltage limit. □

AUTHOR'S BIOGRAPHY

Allan Petersen is a field-applications engineer for Maxim Integrated Products (Hilleroed, Denmark, www.maxim-ic.com). He holds an MSEE from the Technical University of Denmark and has more than 20 years of experience as an R&D engineer and project manager. He enjoys his family, bicycling, traveling, operating HAM radios, and collecting Hewlett-Packard calculators.

Has Your Dream Job Become A Nightmare?



In today's competitive job market, talented electronics engineers and managers have a great opportunity to build a career with a high degree of job satisfaction and life satisfaction.

With EDN's CarEEr Zone, you can do just that. Our sophisticated database technology lets you select the location, position and price range suitable for your skills.

Use keywords to match the right technology, industry or company you want.

And with EDN CarEEr Zone's Personal Search Agent, you can create your dream job and have us e-mail you the matches.

Visit www.ednmag.com and make your dream job a reality.

